

PERANCANGAN APLIKASI SPELL CHECKER DENGAN MENGGUNAKAN ALGORITMA DAMERAU-LEVENSHTEIN

Ferry¹, Riyadi J. Iskandar², Antonius³

^{1,2,3}Teknik Informatika, STMIK Widya Dharma, Pontianak

e-mail: ¹ferry_ichigo92@yahoo.co.id, ²riyadi@stmik-widyadharm.ac.id, ³antoniusok@yahoo.com

Abstract

Spell checker application is an application built that aims to correct the spelling of text that is wrong for the user automatically to save time in case of typing errors. Spell Checker application was designed by using Visual Studio 2012. This study resulted in an Indonesian spell checker application by using algorithms damerau-levenshtein. This application can be run using the Windows 7-based computer that has the specifications intel celeron processor N280, 2.16 GHz, 500GB Hard Drive Storage, Memory (Random Access Memory) 2GB, 64-bit operating system, Windows 7 and .NET Framework v4.5. The authors state that the application can only check the spelling mistakes in Indonesian language. This application also can only inputting documents with the txt extension. By considering the whole process of research, it can be concluded that the spell checker application using damerau-levenshtein algorithm can be used to correct the wrong spelling of text. The suggestions can be given by the author is to add extra function such as autocorrect feature when performing the process of typing, the use of different algorithms to get fewer word suggestions in number but in accordance with the intent of the user.

Keywords—*Damerau, Levenshtein, Spell Checker, Windows 7.*

Abstrak

Aplikasi spell checker adalah sebuah aplikasi yang dibangun yang bertujuan untuk mengoreksi ejaan teks yang salah bagi pengguna secara otomatis untuk menghemat waktu bila terjadi kesalahan pengetikan. Aplikasi Spell Checker ini dirancang dengan menggunakan Visual Studio 2012. Penelitian ini menghasilkan suatu aplikasi spell checker Bahasa Indonesia dengan menggunakan algoritma damerau-levenshtein. Aplikasi ini dapat dijalankan menggunakan perangkat komputer berbasis Windows 7 yang memiliki spesifikasi prosesor intel celeron N280, 2,16 GHz, Harddisk Storage 500GB, Memory (Random Access Memory) 2GB, 64-bit Operating system, dan Windows 7 NET Framework v4.5. Penulis membatasi bahwa aplikasi ini hanya dapat memeriksa kesalahan ejaan Bahasa Indonesia. Aplikasi ini juga hanya dapat meng-input dokumen dengan ekstensi txt. Dengan mempertimbangkan keseluruhan proses penelitian, dapat disimpulkan bahwa aplikasi spell checker dengan menggunakan algoritma damerau-levenshtein dapat digunakan sebagai pengoreksi ejaan teks yang salah. Saran-saran yang dapat diberikan oleh penulis adalah menambahkan fungsi tambahan seperti fitur autocorrect saat melakukan proses pengetikan, penerapan algoritma yang berbeda untuk mendapatkan kata saran yang jumlahnya lebih sedikit namun sesuai dengan maksud user.

Kata kunci—*Damerau, Levenshtein, Spell Checker, Windows 7.*

1. PENDAHULUAN

Penggunaan tata bahasa yang baik sangat berpengaruh dalam menentukan bagus tidaknya tulisan pada sebuah naskah. Kesalahan sekecil apapun akan mempengaruhi maksud yang ingin disampaikan penulis dalam menuangkan idenya. Dengan menggunakan komputer, pengerjaan sebuah naskah akan lebih mudah jika terjadi kesalahan dalam pengetikan. Kesalahan dalam pengetikan tidak dapat dihindari oleh seorang penulis. Pengecekan kesalahan ketik dapat dilakukan dengan memeriksa kembali satu per satu kata yang telah diketik. Namun pemeriksaan kembali memerlukan waktu yang cukup lama dan ketelitian yang tinggi.

Untuk mendapatkan hasil yang benar perlu dilakukan pemeriksaan berkali-kali, proses ini akan sangat melelahkan penulis. Akan lebih efektif jika terdapat sebuah aplikasi pengoreksi ejaan otomatis yang dapat menentukan kata salah ejaan dan memberikan saran kata yang benar bagi penulis.

Penulis menggunakan desain penelitian eksploratif sebagai rancangan penelitian, sedangkan metode pengumpulan data yang digunakan adalah dengan mencari bahan data dan studi kepustakaan. Penulis menggunakan *Unified Modeling Language* sebagai teknik analisis data dalam menggambarkan cara kerja aplikasi. Untuk teknik perancangan sistem, penulis menggunakan aplikasi *Visual Studio 2012* dan *Visual Basic.NET*.

Penelitian yang dilakukan menghasilkan aplikasi *spell checker* Bahasa Indonesia menggunakan algoritma Damerau-Levenshtein.

Kesimpulan yang diperoleh dari perancangan aplikasi *spell checker* menggunakan algoritma damerau-levenshtein adalah aplikasi ini dapat membantu pengguna menghemat waktu untuk proses pengecekan kesalahan ejaan. Untuk lebih mengoptimalkan kinerja aplikasi *spell checker* yang dirancang oleh penulis, diperlukan pengembangan tambahan berupa tambahan pada fitur yang sudah terdapat pada aplikasi dan tambahan fitur-fitur baru lainnya sehingga meningkatkan daya fungsi dari aplikasi *spell checker*.

2. METODE PENELITIAN

2.1. Metodologi Penelitian

2.1.1. Rancangan Penelitian

Rancangan penelitian yang akan digunakan dalam adalah penelitian eksploratif.

2.1.2. Teknik Pengumpulan Data

2.1.2.1. Studi Literatur

Teknik pengumpulan data yang akan digunakan adalah studi literatur yaitu mengumpulkan data melalui referensi dan literatur yang berhubungan dengan aplikasi yang akan dirancang.

2.1.3. Teknik Analisis dan Perancangan Sistem

Teknik analisis dan perancangan sistem yang digunakan penulis dalam penelitian ini adalah *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja dari aplikasi yang dirancang.

2.1.4. Bahasa Pemrograman dan Database

Bahasa Pemrograman yang digunakan penulis dalam merancang aplikasi adalah Microsoft Visual Basic .NET 2012 sebagai sebuah aplikasi pemrograman yang dapat menghasilkan solusi dari permasalahan yang ditulis, sedangkan database yang digunakan adalah SQL Server 2012.

2.2. Landasan Teori

2.2.1. Data

Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. [1]

2.2.2. Informasi

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. [1]

2.2.3. Sistem

Sistem adalah hubungan satu unit dengan unit-unit lainnya yang saling berhubungan satu sama lainnya dan yang tidak dapat dipisahkan serta menuju suatu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan. [2]

2.2.4. Aplikasi

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus). [3]

2.2.5. Perancangan Sistem

Perancangan sistem adalah sebuah teknik pemecah masalah yang saling melengkapi (dengan analisis sistem) yang merangkai kembali bagian-bagian komponen menjadi sistem yang lengkap-harapannya, sebuah sistem yang diperbaikinya. [1]

2.2.6. Perancangan Masukan

Perancangan *input* merupakan *form* pengisian yang berkaitan dengan sistem pengolahan data dalam program. [4]

2.2.7. Perancangan Keluaran

Perancangan *output* merupakan perancangan yang berkaitan dengan informasi hasil pengolahan data. [4]

2.2.8. Perancangan Basis Data

Perancangan basis data mengacu pada aktivitas yang memusatkan pada perancangan struktur basis data yang akan digunakan untuk menyimpan dan mengatur pengguna akhir data. [5]

2.2.9. Damerau-Levenshtein

Sebuah generalisasi yang berbeda dari Levenshtein distance adalah Damerau-Levenshtein distance. Distance ini memungkinkan semua operasi yang diperbolehkan pada Levenshtein distance, dengan tambahan kemampuan transpos dari dua karakter. [6]

2.2.10. Spell Checker

Spell checker adalah sebuah proses mengidentifikasi kata yang memiliki ejaan yang salah dan mengkonversi menjadi kata yang benar. [7]

3. HASIL DAN PEMBAHASAN

3.1. Use Case Diagram Aplikasi Spell Checker

Use Case Diagram (seperti ditunjukkan pada Gambar 1) menjelaskan interaksi antara pengguna dengan sistem aplikasi *spell checker*. Pada saat pengguna mengakses aplikasi, langkah awal yang dilakukan sistem adalah menampilkan menu utama aplikasi *spell checker*. Adapun aksi-aksi yang dapat dilakukan pengguna, di antaranya:

3.1.1. Membuat file baru

Pada form utama, user dapat memilih *new* untuk membuat *file* teks baru dan mengetikkan teks pada *textbox*.

3.1.2. Membuka file teks

User memilih tombol *Open* yang terdapat pada *form* utama dan akan muncul *open file dialog* setelah tombol *Open* dipilih. Pada *open file dialog*, *user* menentukan di mana lokasi *file* teks yang ingin dibuka tersimpan dan memilih *file* teks tersebut. *File* teks yang terpilih akan di-load ke *textbox*.

3.1.3. Memeriksa kata

User memilih tombol *Check* untuk memeriksa jika terdapat kesalahan ejaan. Aplikasi akan membandingkan kata yang di-*input user* dengan kata yang terdapat pada *database* setelah tombol *Check* dipilih. Jika kata yang di-*input* ada kesalahan ejaan atau tidak terdapat di dalam *database*, maka sistem aplikasi akan memberikan kata saran.

3.1.4. Menyimpan file

User dapat menyimpannya pada lokasi penyimpanan yang ditentukan dan memberi nama *file* yang akan disimpan tersebut sesuai keinginan *user*.

3.1.5. Membuat File Baru

User memilih menu *new* dari *form* utama kemudian *user* dapat mengetik teks.

3.1.6. Membuka file teks

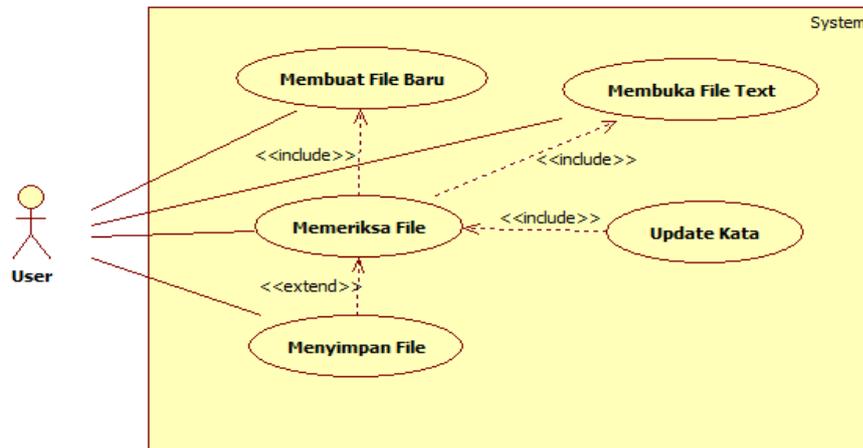
User membuka *sub menu Open* dari *Form* Utama dan memilih lokasi *file* yang ingin diperiksa ejaannya.

3.1.7. Memeriksa kata

Setiap kata akan diperiksa kemiripannya dengan data yang terdapat dalam *database*. Jika kata yang diperiksa terdapat dalam *database* maka kata akan dilewatkan oleh aplikasi, sedangkan jika kata tidak terdapat dalam *database*, maka akan diberikan kata saran dan *user* dapat memilih kata saran yang benar.

3.1.8. Menyimpan file

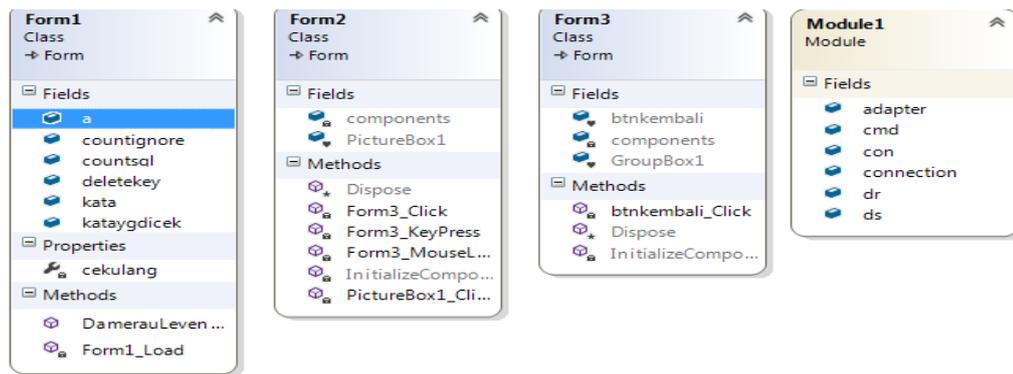
Setelah *file* selesai diperiksa, *user* memilih *Save* dan menentukan lokasi penyimpanan.



Gambar 1 Use Case Diagram Aplikasi Desktop Lock

3.2. Perancangan Database

Berikut ini adalah rincian *class diagram* aplikasi *desktop lock* yang telah dirancang oleh penulis:



Gambar 2 Class Diagram Aplikasi Desktop Lock

3.3. Instalasi Aplikasi

Sebelum menjalankan aplikasi *spell checker* pada *windows*, terlebih dahulu harus meng-*install* aplikasi tersebut pada *windows* yang digunakan. Berikut ini merupakan langkah-langkah proses instalasi aplikasi *spell checker* pada *windows*:

3.3.1. *Copy*-kan *spell checker Bahasa Indonesia.msi*

3.3.2. Jalankan aplikasi *spell checker Bahasa Indonesia.msi*

3.3.3. Proses instalasi aplikasi *spell checker* akan berlangsung selama beberapa detik.

3.3.4. Setelah aplikasi ter-*install* pada komputer maka akan muncul tampilan yang menunjukkan bahwa aplikasi tersebut telah selesai di-*install* dengan baik. Kemudian klik tombol “*Done*”.

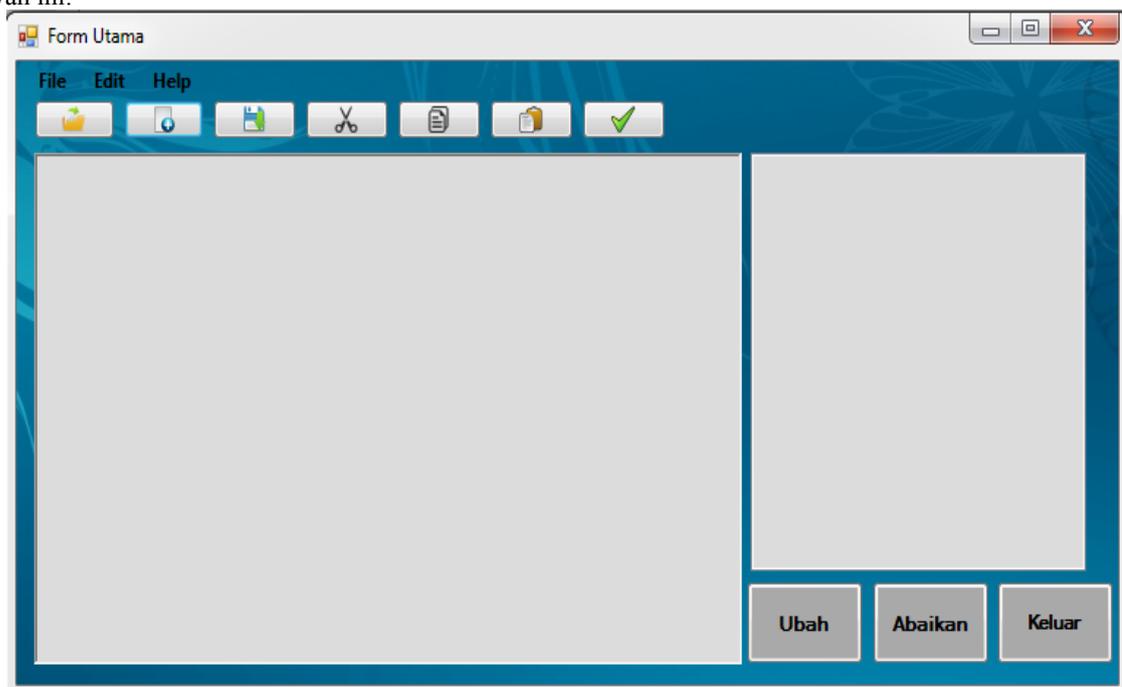
3.4. Tampilan Splash Screen Aplikasi

Pada saat aplikasi dibuka, maka splash screen akan muncul dan tekan key apapun untuk masuk ke form utama.



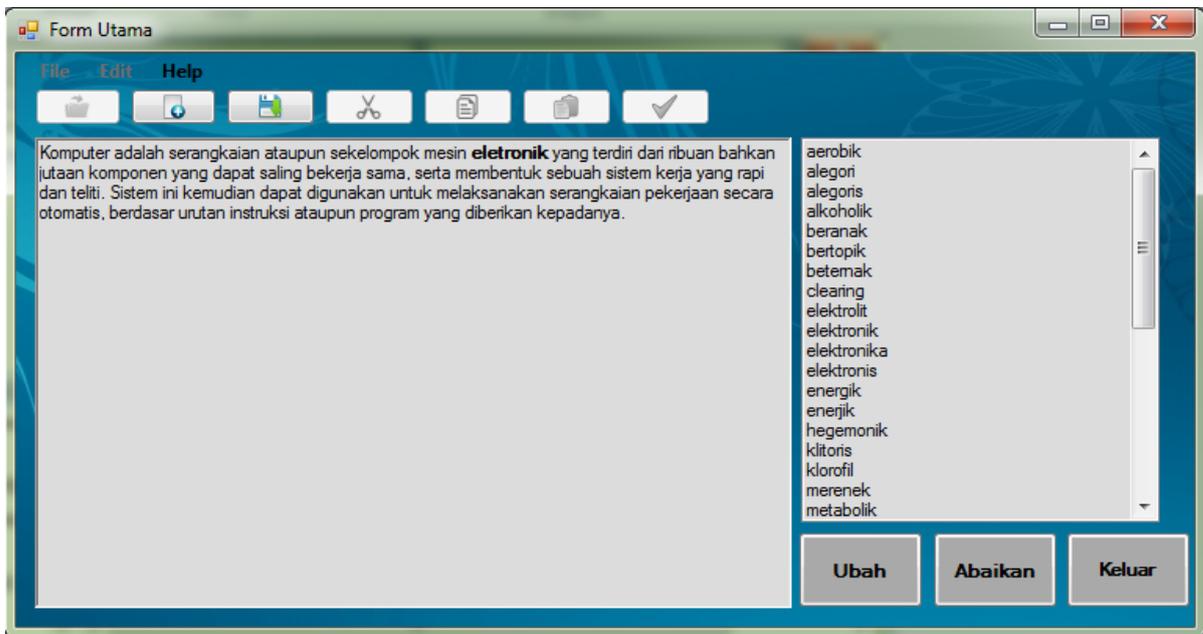
Gambar 3 Tampilan Splash Screen Spell Checker

Pada menu utama terdapat beberapa tombol akses, yakni: tombol “Open”, tombol “New”, tombol “Save”, tombol “Cut”, tombol “Paste”, tombol “Cek”, tombol “Ubah”, tombol “abaikan”, dan tombol “keluar”. Masing-masing dari tombol tersebut memiliki fungsi yang berbeda. Fungsi tombol “Open” adalah untuk Membuka file berekstensi txt untuk diperiksa. Fungsi tombol “new” adalah untuk membuka halaman kosong yang baru. Fungsi dari tombol “save” adalah untuk menyimpan file. Fungsi tombol “Cut” adalah untuk memotong teks yang dipilih. Fungsi tombol “Paste” menempelkan teks yang telah dikopi atau dipotong. Fungsi tombol “Cek” adalah untuk mengecek ejaan yang salah. Fungsi tombol “Ubah” adalah untuk mengganti teks yang salah dengan kata saran. Fungsi tombol “Abaikan” adalah untuk mengabaikan kata saran bila pengguna merasa kata tersebut benar. Fungsi tombol “Keluar” adalah untuk keluar dari aplikasi. Untuk lebih jelasnya dapat dilihat pada gambar 4 di bawah ini:



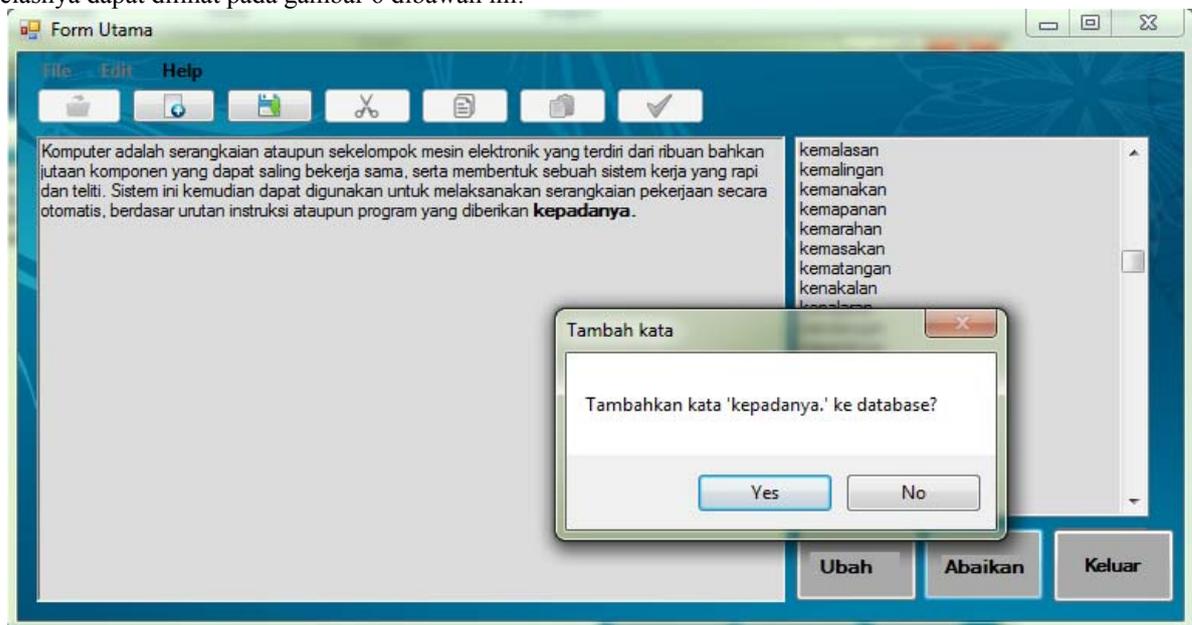
Gambar 4 Tampilan Menu Utama Spell Checker

Pemeriksaan kesalahan ejaan dapat dilakukan setelah teks selesai diketik atau file teks sudah dibuka dan ditampilkan ada textbox. User memilih tombol check untuk melakukan aksi ini. Kata yang memiliki salah ejaan akan dicetak tebal oleh aplikasi. Untuk lebih jelasnya dapat dilihat pada gambar 5 di bawah ini:



Gambar 5 Kata yang salah dicetak tebal oleh aplikasi

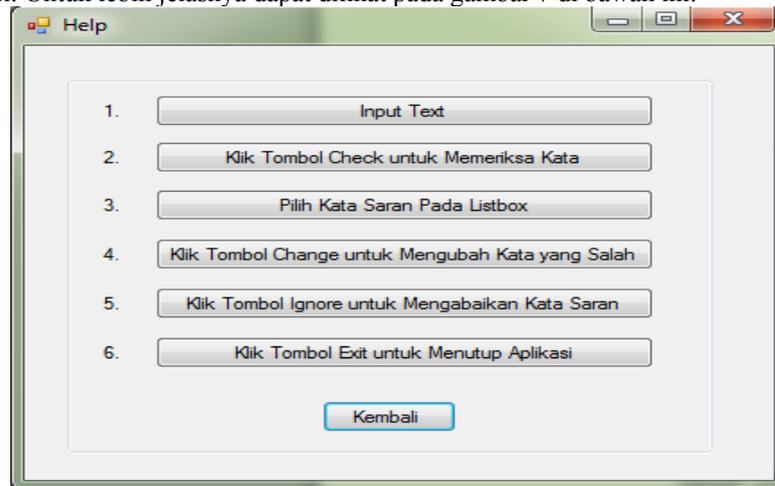
User dapat memilih salah satu kata saran yang terdapat pada listbox dan tombol Change akan aktif secara otomatis. Setelah tombol Change diklik, maka kata “elektronik” di atas diganti menjadi “elektronika”. Apabila didapatkan kata salah ejaan namun dianggap benar oleh user, maka tombol ignore dapat dipilih oleh user dan akan muncul kotak pesan yang menanyakan untuk menyimpan kata tersebut ke database. Untuk lebih jelasnya dapat dilihat pada gambar 6 dibawah ini:



Gambar 6 Kotak pesan tambah kata

Jika kata yang dicek sudah habis, maka akan ditampilkan kotak pesan kepada user bahwa kata sudah habis. Setelah file selesai diperiksa, user dapat menyimpan file dengan memilih tombol save, maka akan ditampilkan kotak dialog menyimpan file dan user dapat memberi nama file tersebut serta memilih lokasi penyimpanan.

Apabila *user* mengalami kendala dalam penggunaan aplikasi, *user* dapat membuka *help* yang menjelaskan cara penggunaan aplikasi. Untuk lebih jelasnya dapat dilihat pada gambar 7 di bawah ini:



Gambar 7 Tampilan Help

1. Pengujian Aplikasi

Pengujian aplikasi bertujuan untuk memastikan bahwa aplikasi yang dibuat penulis berfungsi sesuai dengan yang diharapkan.

Perangkat yang digunakan untuk pengujian aplikasi pemeriksa ejaan Bahasa Indonesia adalah laptop dengan spesifikasi sebagai berikut:

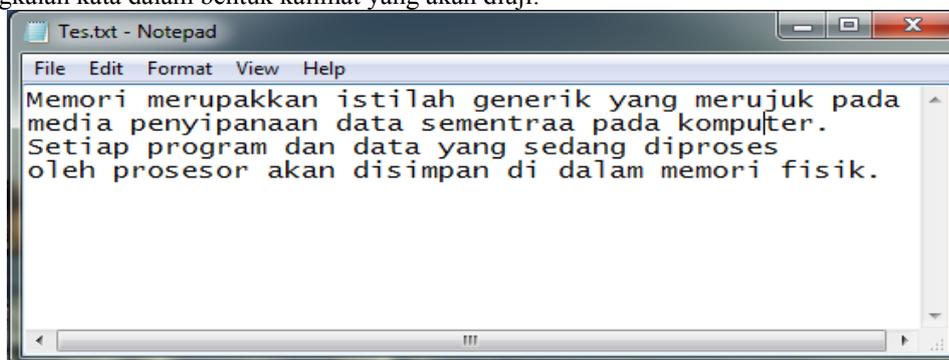
- a. Processor Intel Celeron 2840 @ 2.16 Ghz
- b. Memori RAM 2 GB
- c. Sistem operasi Microsoft Windows 7 Ultimate 64-bit
- d. Hard disk 500GB

Pada proses pengujian menggunakan *file* teks yang telah diketik dan disimpan dalam bentuk *file* berekstensi txt dengan spesifikasi *file* sebagai berikut:

Tabel 1 Spesifikasi File yang Diuji

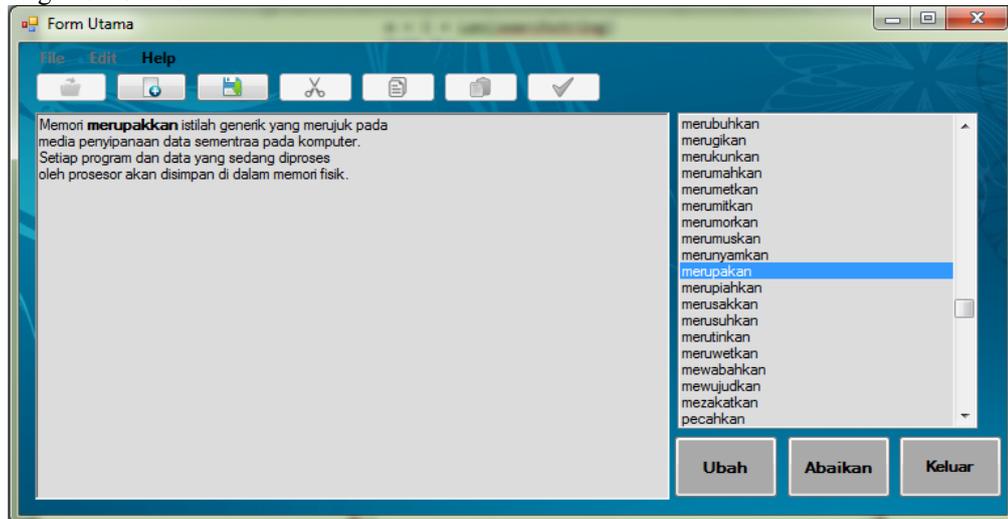
Ukuran File	4 KB
Jumlah Karakter	166
Jumlah Kata	28
Jumlah Kata yang Tidak Sesuai EYD	3
Jumlah Kata yang Sesuai EYD	25

Berikut rangkaian kata dalam bentuk kalimat yang akan diuji:



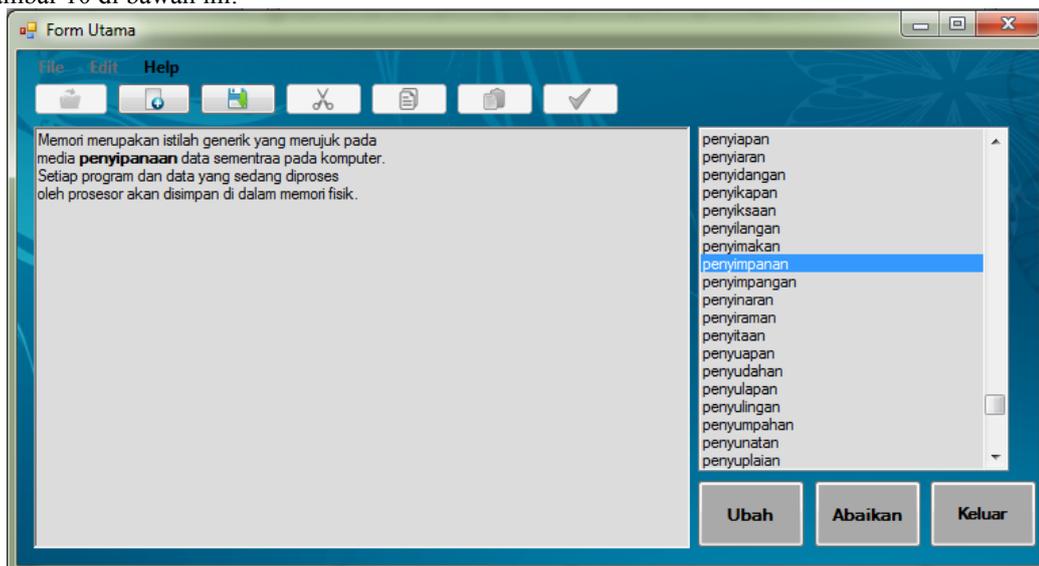
Gambar 8 Kata yang diuji

Pada pengecekan pertama ditemukan kata salah ejaan adalah ‘merupakkan’. Dari hasil pengecekan, didapat hasil yang diinginkan yaitu mengganti kata ‘merupakkan’ menjadi kata ‘merupakan’. Untuk lebih jelasnya dapat dilihat pada gambar 9 di bawah ini:



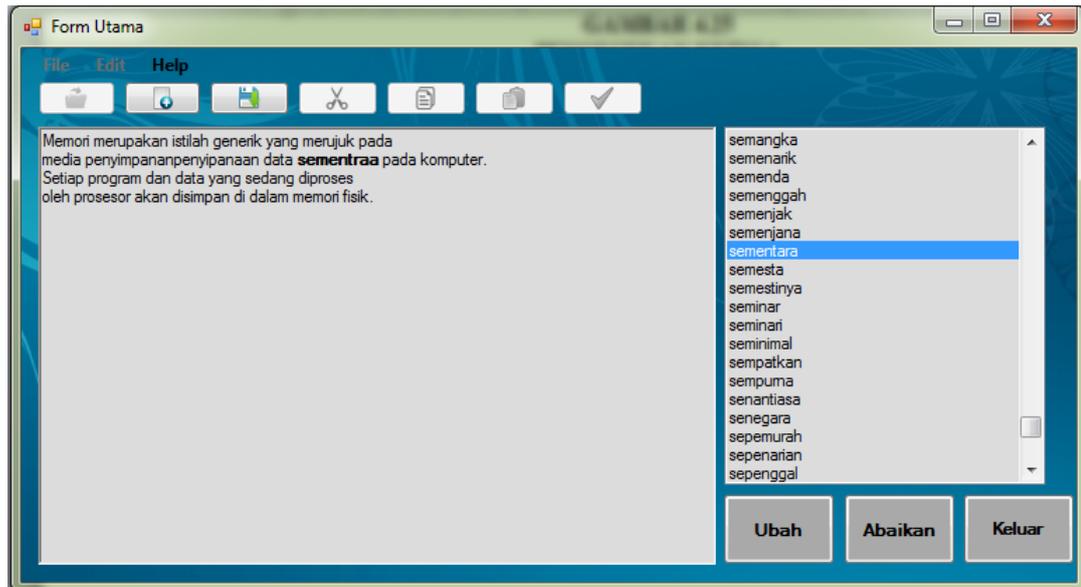
Gambar 9 Pengecekan pertama

Pada pengecekan kedua ditemukan kata salah ejaan adalah ‘penyipanaan’. Dari hasil pengecekan, didapat hasil yang diinginkan yaitu mengganti kata ‘penyipanaan’ mejnjadi ‘penyimpanan’. Untuk lebih jelasnya dapat dilihat pada gambar 10 di bawah ini:



Gambar 10 Pengecekan kedua

Pada pengecekan ketiga ditemukan kata yang salah ejaan adalah ‘sementraa’. Dari hasil pengecekan, didapat hasil yang diinginkan yaitu mengganti kata ‘sementraa’ menjadi ‘sementara’. Untuk lebih jelasnya dapat dilihat pada gambar 11 di bawah ini:



Gambar 11 Pengecekan ketiga

Berdasarkan hasil pengujian yang dilakukan, kata-kata yang salah ejaan dapat ditemukan dan diberikan sejumlah kata-kata saran yang dapat dipilih oleh pengguna. Pada pengujian yang dilakukan terhadap tiga kata yang salah, ketiganya didapatkan kata saran yang diharapkan, sehingga kalimat yang di-input-kan menjadi benar.

4. KESIMPULAN

Berdasarkan hasil penelitian perancangan aplikasi spell checker dengan menggunakan algoritma damerau levenshtein di atas dapat diambil kesimpulan sebagai berikut:

- Aplikasi ini memeriksa kesalahan ejaan dengan membandingkan teks yang di-input user dengan database kata dengan menggunakan algoritma damerau levenshtein distance.
- Jika kata yang di-input user memiliki kesalahan ejaan yang banyak, maka diperlukan juga waktu pengecekan yang lebih banyak, karena aplikasi akan menghitung jarak minimum antar kata yang salah dengan kata pada database satu per satu.
- Format teks pada areainput tidak dapat diformat sesuai keinginan pengguna seperti mengganti warna, ukuran, dan jenis huruf.

5. SARAN

Beberapa masukan untuk pengembangan aplikasi desktop lock pada windows, antara lain:

- Dari sisi user interface masih dibutuhkan perbaikan untuk memudahkan pengguna dalam mengolah kata, seperti fasilitas format ubah warna, ukuran, dan jenis huruf yang ditampilkan pada area input.
- Jumlah kata dalam database masih terlalu sedikit sehingga perlu ditambahkan database kata dalam jumlah besar agar akurasi pengecekan dokumen lebih akurat.
- Penerapan algoritma yang berbeda untuk mendapatkan kata saran yang jumlahnya lebih sedikit, namun sesuai dengan yang dimaksud oleh user.
- Aplikasi men-support untuk membuka file dokumen dengan ekstensi lain selain .txt.
- Penambahan fitur autocorrect saat melakukan pengetikan.
- Penambahan fitur untuk memeriksa tata bahasa.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada orang tua, dosen, dan teman-teman yang telah memberi dukungan dan masukan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Al-Fatta, Hanif. (2007). Analisis & Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan & Organisasi Modern, edisi pertama. C.V ANDI OFFSET. Yogyakarta.
- [2] Gaol, Chr. Jimmy L. (2008). Sistem Informasi Manajemen Pemahaman dan Aplikasi. PT GRASINDO. Jakarta.
- [3] Hendrayudi. (2009). VB 2008 untuk Berbagai Keperluan Pemrograman. PT Elex Media Komputindo. Jakarta.
- [4] Kursini, dan Andri Koniyo. (2007). Tuntunan Praktis Membangun Sistem Informasi Akuntansi dengan Visual Basic dan Microsoft SQL Server. C.V ANDI OFFSET. Yogyakarta.
- [5] Simarmata, Janner. (2007). Perancangan Basis Data. C.V ANDI OFFSET. Yogyakarta.
- [6] Raś, Zbigniew W. dan Alicja A. Wiczorkowska. (2010). Advances in Music Information Retrieval. Springer-Verlag. Berlin.
- [7] Zhou, Shuigeng dan Zhiang Wu. (2012). Social Media Retrieval and Mining. Springer. London.